⑲日本国特許庁（JP）　⑪特許出願公開

⑫公開特許公報（A）　昭64－68835

⑤1Int.Cl.⁴	識別記号	庁内整理番号	⑬公開　昭和64年(1989)3月14日
G 06 F 9/06	330	A－7361－5B	

審査請求　未請求　発明の数 1　(全13頁)

⑤4発明の名称　ソフトウエア権利管理制御方法

⑵1特　願　昭62－227106

⑵2出　願　昭62(1987)9月10日

⑦2発明者　森　亮一　東京都文京区白山1－24－12

⑦2発明者　田代　秀一　千葉県柏市松葉町1－12－20－2

⑦1出願人　森　亮一　東京都文京区白山1－24－12

⑦4代理人　弁理士　小笠原　吉義　外2名

明　細　書

1．発明の名称　ソフトウェア権利管理制御方法

2．特許請求の範囲

（1）暗号化された命令を，命令フェッチ時に復号化して実行するデータ処理装置におけるソフトウェア権利管理制御方法であって，

権利管理状態の変更に関する割込みを制御する権利管理割込み発生機構(13)を設け，

権利管理対象となるプログラムまたはプログラム群に対し，暗号化された命令を復号化する鍵を個別に割り当て，

上記権利管理割込みにより，上記復号化する鍵を切り換える(14)ことを特徴とするソフトウェア権利管理制御方法。

（2）上記権利管理対象となるプログラムまたはプログラム群は，許諾条件プログラム，該許諾条件プログラムによって実行を制御されるソフトウェア本体，またはそれらの組み合わせであることを特徴とする特許請求の範囲第（1）項記載のソフトウェア権利管理制御方法。

3．発明の詳細な説明

〔概要〕

個々の権利対象ソフトウェア毎に，ソフトウェアの使用許諾，使用記録情報の収集などを，他のソフトウェアによる影響および干渉を受けずに行うことができるようにしたデータ処理装置におけるソフトウェア権利管理制御方法に関し，

マルチプログラミングの環境における権利管理の実現に適した制御方法を提供することを目的とし，

暗号化された命令を，命令フェッチ時に復号化して実行するデータ処理装置におけるソフトウェア権利管理制御方法であって，権利管理状態の変更に関する割込みを制御する権利管理割込み発生機構を設け，権利管理対象となるプログラムまたはプログラム群に対し，暗号化された命令を復号化する鍵を個別に割り当て，上記権利管理割込み

により，上記復号化する鍵を切り換えるように構成する。

〔産業上の利用分野〕

本発明は，個々の権利対象ソフトウェア毎に，ソフトウェアの使用許諾，使用記録情報の収集などを，他のソフトウェアによる影響および干渉を受けずに行うことができるようにしたデータ処理装置におけるソフトウェア権利管理制御方法に関する。

半導体技術の進歩によって，計算機ハードウェアの価格低下，小型化，高機能化が急速に進みつつある。ハードウェア機能の高度化は，ソフトウェアの高度化を要求し，結果として，ハードウェア開発者，ソフトウェア開発者，利用者の明確な分業化が促進されている。

しかしながら，ソフトウェア流通の環境は，現在，混乱した状況にあるといってよく，その環境を改善するために，ソフトウェア権利者の保護と利用者の便利さとを両立させるシステムであって，

ソフトウェアの自由な大量流通を支援するシステムを構築することが望まれている。

〔従来の技術〕

ソフトウェアの開発には，大量の人員および時間を要するにもかかわらず，利用者または第三者は，完成したソフトウェアを複製して，全く同じものを作成することが比較的容易にできる。そのため，ソフトウェアを開発した権利者（ソフトウェア権利者）は，開発コストの回収が不能になるケースが多く，これに対して，(1)ソフトウェア製品の価格を盗用による損失分を上乗せした値とする，(2)契約によって無断複製を禁止する，(3)ソフトウェアを複製不能な媒体に封入して流通させる，(4)計算機のシリアルナンバーなどにより使用可能な装置を限定する，などによって対処することが行われている。

しかしながら，上記(1)は，正当な利用者に過度の負担を与え，上記(2)は，繁雑な契約手続きが必要になると共に，それによる効果が充分でなく，

上記(3)は，バックアップがとれなくなるという問題や，複製不能にしている手段を調べ，それを回避することにより複製可能となるという問題があり，上記(4)は，特定計算機を識別するために，ソフトウェアの販売時などに特殊化処理が必要となり，自由なソフトウェアの流通が阻害されるという問題がある。

これらの問題を解決するものとして，本出願人は，ソフトウェア権利者の権利と，利用者の自由と，流通の容易性の3つを同時に満たすソフトウェアサービスシステム（ＳＳＳ）の基本構想を提案している。〔森亮一："ソフトウェアサービスシステム"，電子通信学会誌，Vol.67，No.4，pp.431-436（Apr.1984）〕

ここで提案されたソフトウェアサービスシステムの基本は，

(1) ソフトウェアは，いかなる条件が成立した場合に，その実行を許すかの許諾条件を内部に持つべきである。

(2) 計算機内部には，利用者の持つ権利を記述した権利記録があるべきである。これには，共通クレジット，買い取り記録，試用記録，特別許諾コード等が有り得る。

(3) 計算機は，上記(2)の権利記録が上記(1)の許諾条件を満足した場合にのみ，ソフトウェアの利用を許す実行管理機構を持つべきである。

(4) 権利者が，ソフトウェアの利用状況を把握できるように，計算機は，回収用作業記録を持つべきである。利用者に特別な面倒を与えることなく，この作業記録を回収する電子的手段が有り得る。

…というものである。

ところで，このソフトウェアサービスシステムの提案とは別に，暗号化された命令およびデータを主記憶にロードし，命令フェッチ時に，計算機のシリアルナンバーなどの復号鍵によって復号して実行する計算機アーキテクチャが知られている。

〔発明が解決しようとする問題点〕

上記ソフトウェアサービスシステムを，マルチプログラミングの環境下で実現する場合，複数の

ソフトウェア権利者に係る複数のプログラムが，時分割的に同時に計算機資源を利用して，走行することになる。この場合，第１に，権利の管理に係るハードウェア機構を制御する主体を，いかにして許諾条件をチェックするプログラム（以下，許諾条件プログラムという）に限定するか，第２に，マルチプログラミングの環境において，許諾条件プログラムと，それによって実行を制御されるソフトウェア本体との対応関係をいかにして保つか，第３に，オペレーティング・システムに対し，スワッピングやプロセス・スイッチを行うためのアクセス権を認めつつ，許諾条件プログラムの内容を改ざんしたり，権利管理のための流れを乱したりすることがないように，いかにしてアクセス権を制限するか，といったことに対する考慮が重要となる。従来，このような権利管理に対する適切な計算機の実行制御を行う機構はなかった。

本発明は上記問題点の解決を図り，マルチプログラミングの環境における権利管理の実現に適した制御方法を提供することを目的としている。

〔問題点を解決するための手段〕

第１図は本発明の原理説明図である。

第１図において，１０は権利管理の単位となる権利管理対象プログラム，１１は許諾条件をチェックしソフトウェア本体の実行を制御する許諾条件プログラム，１２はワープロソフト，コンパイラなどの利用者の使用目的に応じた処理を行うソフトウェア本体，１３は権利管理状態の変更に関する割込みを制御する権利管理割込み発生機構，１４は権利管理割込みによって権利管理状態を変更する権利管理状態の変更処理，１５はプログラムステータスワード（ＰＳＷ），１６は権利管理対象プログラム１０を識別する権利管理ＩＤ（ＰＭＩＤ），１７は権利管理対象プログラム１０の暗号化された命令を復号化する鍵を権利管理ＩＤ１６毎に記憶管理する権利管理テーブル，１８は暗号化された命令をその命令のフェッチ時に復号化する復号化回路を表す。

権利管理対象プログラム１０は，例えば許諾条

件プログラム１１とソフトウェア本体１２との対によって構成される。これらの全部または一部は，権利管理対象プログラム１０毎に定められた１または複数の暗号鍵によって予め暗号化されている。

権利管理対象プログラム１０のロード時に，暗号化された命令を復号化する鍵情報が，その権利管理対象プログラム１０に対してシステムでユニークに割り当てた権利管理ＩＤ１６と共に，権利管理テーブル１７に書き込まれる。

プログラムステータスワード１５は，カレントの権利管理ＩＤ１６が設定されるフィールドを持っている。プログラム実行時に，プログラムステータスワード１５に設定されている権利管理ＩＤ１６により，権利管理テーブル１７の該当エントリがアクセスされ，そのエントリ中の復号鍵（ＫＥＹ）が読み出されて，復号化回路１８へ送られる。復号化回路１８は，暗号化された命令をフェッチしたときに，権利管理テーブル１７から読みだした復号鍵によって復号し，その復号した命令を計算機の命令実行ユニットへ送る。

本発明では，オペレーティング・システム等の処理機能を起動するスーパバイザ割込みなどとは別に，権利管理状態に関する変更を制御する権利管理割込み発生機構１３を設ける。そして，権利管理割込み発生機構１３により，権利管理割込みが発生した場合に，権利管理状態の変更処理１４によって，権利管理テーブル１７への権利管理状態の退避やプログラムステータスワード１５のカレントの権利管理ＩＤ１６の変更などを行う。これにより，命令などを復号化する鍵を切り換える。

〔作用〕

本発明によれば，割込みが，オペレーティング・システムによる資源管理などに関連する従来の割込みと，権利管理状態を制御するための権利管理割込みの２系統に分離されることになる。

権利管理割込みが発生したときに，権利管理状態の変更処理１４によって，権利管理テーブル１７に基づく権利管理状態を変更するので，権利管理に係るハードウェア機構を制御することができ

る主体を，特定の許諾条件プログラム１１などに個別的に限定することができる。また，個々に復号鍵が管理されるので，ある許諾条件プログラム１１に対応した権利管理テーブル１７中のエントリ等，権利管理に関する情報を，他の権利管理対象プログラム１０などによって，改ざんされることを防止することができる。

また，割込みが，少なくとも従来の割込みと権利管理割込みの２系統になるので，オペレーティング・システムによる権利管理に対する望ましくない関与も，防止することが可能となる。

〔実施例〕

第２図は本発明を用いたソフトウェアの流通形態の例を説明するための図，第３図は本発明による権利管理の説明図，第４図は権利管理状態の状態遷移図，第５図は本発明に用いるハードウェア構成例，第６図は権利管理対象となるソフトウェアの構成例，第７図は権利管理テーブルの構成例，第８図はＰ割込み要求テーブルの構成例，第９図

はＳ割込み制御の例，第１０図はＳ割込みからの復帰制御の例，第１１図は本発明の一実施例におけるスタックの使用例，第１２図は本発明の一実施例における空間と鍵の関係説明図を示す。

本発明は，ソフトウェアの保護と利用促進のため，例えば第２図に示すようなソフトウェアの流通形態を採用する場合に用いることができる。

① ソフトウェアを開発したソフトウェア権利者は，そのソフトウェアを流通させる場合，ソフトウェアの利用者に対して要求する条件を記述した許諾条件プログラムを作成し，それを開発したソフトウェア本体と結合して，暗号化し，電波による放送も含めた自由な媒体により，ソフトウェアを配布する。

② ソフトウェアを利用したい利用者は，例えば予め共通クレジット（ＣＣ）を自動販売機などのベンディングマシンにより購入する。この共通クレジットの内容は，例えばＩＣカードによる媒体を介して，データ処理装置２０に入力できるようになっている。

③ 利用者は，共通クレジットの内容が記録されたＩＣカードを，本発明に係るソフトウェア権利管理制御機能を持つデータ処理装置２０に装着し，適当な流通路から入手したソフトウェアを，データ処理装置２０上で動作させる。これにより，そのソフトウェアの許諾条件プログラムが復号化されて動作し，ＩＣカードへのアクセスにより，共通クレジット情報に関連するチェックなどを行う。必要に応じて共通クレジットに記録されている料金の減算を行う。チェックに合格した場合にのみ，必要なソフトウェア本体の実行を可能とする制御を行う。そのとき，その利用に関する作業記録情報を蓄積しておき，その作業記録（ＡＲ）を適当な時期にＩＣカードへ転記する。

④ ＩＣカードに書き込まれた作業記録は，例えば共通クレジットを継続使用するために，新たな料金の支払いにより共通クレジットの内容を再設定するときに，ベンディングマシンに読み取られ，収集される。この作業記録情報を参照することにより，ソフトウェア利用状況の統計をとり，それに基づいて複数のソフトウェア権利者間で妥当な料金の分配を行うことができる。

第２図に示したソフトウェア流通形態は，一例であり，本発明は，共通クレジットではなく，各ソフトウェア個別の利用資格情報を管理する流通形態をとる場合にも採用することができる。また，ＩＣカード以外の媒体を用いることも可能である。

本発明による場合，データ処理装置２０は，複数の独立したソフトウェアを，マルチプログラミングによって並列動作させることができる。

本発明による権利管理は，概念的には，第３図に示すようになる。

マルチプログラミングの環境では，オペレーティング・システムが，入出力装置の割り当て，プロセッサの割り当てなどの計算機ハードウェア資源の管理を行っている。この場合，誤ったプログラムの実行などによって支障をきたさないようにするために，また計算機の資源を多く利用しようとする利用者によって，資源が恣意的に用いられることを防ぐために，計算機の実行状態をユーザ

状態と，スーパバイザ状態に分けることが広く行われている。

利用者のプログラムは，ユーザ状態で実行され，資源管理用のオペレーティング・システムは，通常，スーパバイザ状態で実行される。そして，利用者による資源の濫用を防止し，また利用者が勝手に実行状態を変更できないようにするために，ユーザ状態からスーパバイザ状態へと状態を変更する手段を，割込み（割出しを含む）に限ることによって，両状態の分離を図っている。

ここで，第２図で説明したような許諾条件プログラムの復号化や，共通クレジットの読み書きなどに関連する権利管理を，スーパバイザ状態のもとで，オペレーティング・システムにより制御するとすれば，利用者が自己の都合のよいように内部を改変したオペレーティング・システムをロードすることに関して，抵抗力を持たない。これに対し，オペレーティング・システムをファームウェア化して，内容の変更を困難にするという対応策も考えられるが，オペレーティング・システム

のバージョンアップが不可能になるなどの他の問題が発生する。

そこで本発明では，スーパバイザ状態とは異なる権利管理状態という特別な実行状態を新設している。権利管理状態は，ソフトウェアをできるだけ無料で実行したいと考える利用者があり得る環境の中で，利用者の故意または過失による介入を避けつつ，外部から供給されたソフトウェアの権利に関する管理を行うためのプログラムを実行可能とする状態である。

権利管理状態は，他の状態からそこへ状態を変更する手段を割込みに限っている点および他の状態では実行できない命令を持つ点で，スーパバイザ状態に類似しているが，暗号の応用と鍵の管理とによって不正なアクセスの防止を図る点が，スーパバイザ状態と大きく異なる。

第３図に示すように，オペレーティング・システムが，スーパバイザ状態のもとで，資源管理を行うのに対し，権利管理対象となっているプログラムの各許諾条件プログラム１１Ａ，１１Ｂ，…

は，個々の権利管理状態のもとで，ソフトウェア本体１２Ａ，１２Ｂ，…の実行を監視することによって，権利管理を行う。以下，許諾条件プログラムと，それによって制御されるソフトウェア本体とを合わせて権利管理対という。

権利管理状態を設けることにより，本発明に係るデータ処理装置は，例えば第４図に示すような状態遷移を行う。ユーザ（ＵＳ）状態，スーパバイザ（ＳＶ）状態，権利管理（ＰＭ）状態があり，スーパバイザ状態は，さらにＳＶｐ状態とＳＶｕ状態とに分かれる。ＳＶｐ状態とＳＶｕ状態とは，使用できる命令，アクセス権等に関しては全く同様であるが，割込みからの復帰命令を実行した場合に，ＰＭ状態に復帰するかＵＳ状態に復帰するかの点が異なる。

以上の４つの状態（ＳＶｐ，ＳＶｕ，ＰＭ，ＵＳ）は，独立した２つの状態ビットにより表現される。その１つは，プログラムステータスワード１５中に置くＳビットであり，他の１つは権利管理テーブル１７中に置くＰビットである。

Ｓビットの変化に伴う状態遷移は，スーパバイザ状態で実行するプログラム，即ち，オペレーティング・システムの制御に関係するものであり，Ｐビットの変化に伴う状態遷移は，ＰＭ状態で実行されるプログラム，即ち，許諾条件プログラムの制御に関係するものである。

これら２系統の状態遷移を直交した関係におき，互いに他の系統の状態遷移には直接影響を及ぼさないようにしているので，資源管理と権利管理との制御機構の分離が図られている。

以下，Ｓビットに関連する従来の割込みをＳ割込みといい，Ｐビットに関連する本発明に係る権利管理割込みをＰ割込みという。

本発明に用いるハードウェア構成は，例えば第５図に示すようになっている。

第５図において，第１図と同符号のものは，第１図に示すものに対応する。３０は命令実行ユニットである演算／制御部，３１はＩＣカードインタフェース，３２はＩＣカードから入力された共通クレジット情報を記憶する共通クレジットレジ

スタ（ＣＣＲ），３３はソフトウェア買い取り情報などの個々のソフトウェアに特有な権利管理情報を持つ個別権利メモリ（ＳＲＭ），３４はソフトウェアの利用実績情報が記録される作業記録メモリ（ＡＲＭ）である。

共通クレジットレジスタ３２，個別権利メモリ３３，作業記録メモリ３４は，不揮発メモリによって構成される。

３５は許諾条件プログラムまたはソフトウェア本体の全部または一部をＤＥＳ方式により復号化するＤＥＳ暗号機構，３６はＰ割込みの要因などの制御情報を記憶するＰ割込み要求テーブル（ＰＩＴ），３７は権利管理対象となるソフトウェアをロードする際に後述するキー格納部の復号化を行う公開鍵暗号機構である。

第５図に示す装置が扱うソフトウェアの構造は，例えば第６図に示すようになっている。キー格納部，許諾条件プログラム１１，ソフトウェア本体１２の３つの部分からなる。

本実施例では，許諾条件プログラム１１を，権

利者が任意に与える鍵（ＫＥＹ１）によって，いわゆるＤＥＳ方式で暗号化する。また，ソフトウェア本体１２の少なくとも一部を，やはり権利者が任意に与える鍵（ＫＥＹ２）によってＤＥＳ方式で暗号化する。なお，ＤＥＳ方式については，例えば「一松信監修："データ保護と暗号化の研究"，日本経済新聞社〔1983〕」に記述されている。

キー格納部には，このソフトウェアを識別するユニークな権利番号（ＰＮ）と，個別権利記録（ＳＲ）に対するアクセスを管理するための非公開のＳＲアクセスキーと，許諾条件プログラム１１，ソフトウェア本体１２を復号化するためのＫＥＹ１，ＫＥＹ２とが設定されるようになっている。キー格納部は，例えば公開鍵暗号方式のひとつであるＲＳＡ法により暗号化される。なお，このＲＳＡ法も，例えば上記"データ保護と暗号化の研究"の著書などにより知られている。公開鍵暗号方式は，暗号化に用いる鍵を知っていても，秘密鍵である復号化鍵を知っていないと，暗号を

解くことができないという特徴がある。もちろん他の暗号方式を用いても，同様に本発明を実施することは可能である。

このキー格納部を復号化するための鍵は，ハードウェア製造時に，第５図に示す公開鍵暗号機構３７の内部に封入しておく。このソフトウェアを主記憶上にロードする際に，公開鍵暗号機構３７によってキー格納部を復号化し，その復号結果を権利管理テーブル１７に設定する。

権利管理テーブル１７は，例えば第７図に示すような権利管理に関する情報を持つ。

権利管理テーブル１７中のレコードは，このレコードに対応する権利管理対を識別するためのＰＭＩＤ，権利番号（ＰＮ），ＳＲアクセスキー，許諾条件プログラムを実行するために必要なＤＥＳ鍵（ＫＥＹ１），ソフトウェア本体を実行するために必要なＤＥＳ鍵（ＫＥＹ２），ＫＥＹ２の有効／無効を示すビット（Ｋ２Ｆ），許諾条件プログラムによる権利管理状態か否かを表すＰビット，対応する権利管理対がロードされた直後であ

ることを表すイニシャルビット，およびスタックの正当性を確認するためのスタックチェックコードの９つのフィールドからなる。

システム中に同時に存在し得る権利管理対の数は，権利管理テーブル１７のレコード数に等しい。複数のレコードのうち，ＰＳＷ中のカレントＰＭＩＤの値と一致したＰＭＩＤを持つレコードを，カレントＰＭＴレコードと呼び，権利管理テーブル１７のレコードに対するアクセスは，通常の場合，このカレントＰＭＴレコードに対して行われる。

第５図に示すＰ割込み要求テーブル（ＰＩＴ）３６は，権利管理割込み（Ｐ割込み）発生機構１３が使用するテーブルであり，例えば第８図に示すような構成になっている。ＰＩＴ３６中の各レコードは，Ｐ割込みを要求する権利管理対のＰＭＩＤ，そのＳＲアクセスキーを記憶するフィールド，要求するＰ割込みの要因を記述するフィールド（Ｐ－Reason），Ｐ割込み発生時の飛び先アドレスを記述するフィールド（New-PC），Ｐ割込み

要求の要因に対するパラメータを記述するフィールドの５つのフィールドからなる。

ＰＩＴ３６中のレコードは，ＰＭ状態でしか使用できない特権命令によってのみ，生成／消去が可能にされ，また，ＳＲアクセスキーを用いたアクセス権のチェックにより，正当な権利者以外の許諾条件プログラムからは，既に存在するＰＩＴレコードの書き換えや消去ができないようになっている。

Ｐ割込みの要因には，例えばＵＳ状態におけるＰ割込み命令の実行，タイマー割込み，ＵＳ状態における特定命令の実行（命令トラップ）がある。タイマー割込みに対しては，パラメータとして割込み発生の時間間隔を与えることができ，命令トラップに対しては，任意の命令コードをパラメータとして与えることができる。これにより，許諾条件プログラムは，必要に応じてソフトウェア本体の実行制御を行い，またソフトウェア本体の実行に関する作業記録をとる契機を得ることができるようになっている。

トＰＭＴレコードから取り出した旧スタックチェックコード１，２の４ワード（６４bit）をＫＥＹ１で暗号化し，スタックへストアする。

(e) 新スタックチェックコード１，２を，カレントＰＭＴレコードへストアする。

(f) ＰＩＴの飛び先アドレス（New-PC）を，プログラムカウンタにセットする。そして，(h)の制御へ移る。

(g) Ｐビットが０の場合には，割込みベクタをプログラムカウンタにセットする。

(h) ＰＳＷ，プログラムカウンタ，スタックフォーマットコードをスタックへストアする。

(i) ＰＳＷのＳビットを"１"にし，オペレーティング・システムの割込み処理ルーチンへ制御を移す。

なお，この例では，Ｓ割込み時における飛び先アドレス，即ち，プログラムカウンタに設定する新しい値を，いわゆる割込みベクタテーブルからロードする方式を用いている。

スーパバイザ状態において，ＲＳＥ（Return from S-Exception）命令を実行することにより，ＰＭ状態またはＵＳ状態のいずれかへ遷移する。そのときのプロセッサの動作は，例えば第１０図に示す(a)～(g)のようになる。

第５図に示すＰ割込み発生機構１３は，許諾条件プログラムがＰＩＴ３６に指定した要因が発生したときに，自動的にＰ割込みを起こす機構であるが，その内部の回路構成等については，従来技術と同様な一般的な割込み技術を用いて実施可能であるので，これ以上の説明を省略する。

Ｓ割込みとＰ割込みとを独立させるために，Ｓ割込みが発生した場合におけるプロセッサの動作は，例えば第９図に示す(a)～(i)のようになる。

(a) Ｓ割込みが発生した場合，ファームウェアなどにより，カレントＰＭＴレコードのＰビットの０／１をチェックし，０であれば(g)へ制御を移す。

(b) Ｐビットが１であれば，現在のＰＳＷ及びプログラムカウンタの値から，１６bit のサイクリックコードを生成し，新スタックチェックコード１とする。

(c) １６bit の乱数を生成し，新スタックチェックコード２とする。

(d) 新スタックチェックコード１，２及びカレントＰＭＴレコードから取り出した旧スタックチェックコード１，２の４ワード（６４bit）をＫＥＹ１で暗号化し，スタックへストアする。

(e) 新スタックチェックコード１，２を，カレントＰＭＴレコードへストアする。

(f) ＰＩＴの飛び先アドレス（New-PC）を，プログラムカウンタにセットする。そして，(h)の制御へ移る。

(g) Ｐビットが０の場合には，割込みベクタをプログラムカウンタにセットする。

(h) ＰＳＷ，プログラムカウンタ，スタックフォーマットコードをスタックへストアする。

(i) ＰＳＷのＳビットを"１"にし，オペレーティング・システムの割込み処理ルーチンへ制御を移す。

(a) カレントＰＭＴレコードのＰビットの０／１をチェックする。それが０であれば，(f)へ制御を移す。

(b) スタックから，スタックチェックコード１，２および旧スタックチェックコード１，２を取り出し，それらをＫＥＹ１で復号化してプロセッサ内に保持する。

(c) スタックのＰＳＷ及びプログラムカウンタの値から，１６bit のサイクリックコードを生成し，上記(b)で取り出したスタックチェックコード１と比較する。不一致であれば，エラーとして処理する。即ち，ＰＭ状態への遷移などを防止する。

(d) 次に，カレントＰＭＴレコードのスタックチェックコード２と，上記(b)で取り出したスタックチェックコード２とを比較する。不一致であ

ればエラーとして処理する。

(e) 旧スタックチェックコード1，2をカレントＰＭＴレコードへストアする。

(f) ＰＳＷのＳビットを"０"に戻す。

(g) スタックにストアされているプログラムカウンタの値を，プログラムカウンタにセットする。そして，割込み発生直前に実行されていたルーチンへ復帰する。

ＰＭ状態からＳ割込みによりＳＶp 状態へ遷移する際に，例えば第１１図（ａ）に示すようなスタック情報が，プロセッサによってスーパバイザ空間に退避される。スタックチェックコードは，スタックに退避された情報が，オペレーティング・システムによって書き換えられることを防止するために使用されるコードである。

スタックチェックコード1は，第９図で説明したように，スタックに退避されるＰＳＷ及びプログラムカウンタの値から生成するサイクリックコードであり，スタックチェックコード2は，１６bit の乱数である。これらは，スタックに退避さ

れると同時に，カレントＰＭＴレコードにも格納され，後にスタックの情報を取り出す際にチェックされる。これにより，スタック内容が書き換えられたことを検出できるようになっている。

旧スタックチェックコード１，２は，割込みがネストされた場合に，古いスタックチェックコードを記録するためのものである。

ＵＳ状態からＳＶu 状態へ遷移する場合には，第１１図（ｂ）に示すデータがスタックに退避される。また，ＳＶ状態時におけるＳ割込み発生時には，第１１図（ｃ）に示すデータがスタックに退避され，ＵＳ状態時におけるＰ割込み発生時には，第１１図（ｄ）に示すデータがスタックに退避される。なお，ＳＶ状態においては，Ｐ割込みはマスクされる。

本実施例では，ＳＶ，ＰＭ，ＵＳの各状態毎に，プログラムのアクセスできる空間およびハードウェアによって自動的に使用される暗号の鍵が決められる。第１２図は，その空間と鍵の関係を示している。

ＰＭ状態におけるプログラムの実行は，以下のとおりである。

ＰＭ状態において，命令フェッチ，データのロード／ストア，サブルーチン呼び出し，割込み発生時におけるプログラムカウンタのスタックへの退避など，主記憶に対するあらゆる書き込み，読み出しの際に，カレントＰＭＴレコード中のＫＥＹ１を用いた自動的な暗号化および復号化が，第５図に示すＤＥＳ暗号機構３５によって行われる。ＰＭ状態で使用するアドレス空間は，ＵＳ状態の空間と全く同一であり，区別されてない。これらは，鍵の管理によってのみ隔離されている。

オペレーティング・システムといえども，鍵を知らない限り，許諾条件プログラムを一時停止，強制終了させること以外，その動作に不正に介入することはできない。

ＰＭ状態からＵＳ状態への戻りは，ＲＰＥ（Return from P-Exception)命令の実行によって行われる。

ＵＳ状態においては，命令フェッチ時に限り，

カレントＰＭＴレコードのＫＥＹ２を用いた復号化を行う。ただし，復号化による命令実行速度の低下を軽減するために，プログラムのすべての部分を暗号化するのではなく，暗号化する範囲を任意に指定できるようにする方式をとることが望ましい。そのため，ＵＳ状態で，カレントＰＭＴレコードのＫ２Ｆビットをコントロールする命令が用意され，Ｋ２Ｆビットが"１"の間だけ，命令フェッチ時に，ＫＥＹ２による復号化を自動的に行うようになっている。

オペレーティング・システムは，以下の処理を行う。

① 権利管理対象のソフトウェア全体を，暗号化されたまま主記憶へロードする。

② 次に，権利管理テーブル１７内に新しいレコードを生成する命令を発行する。この命令では，これから実行しようとする権利管理対に与えるＰＭＩＤの値と，ロードしたソフトウェアのキー格納部の先頭論理アドレスとをパラメータとする。ＰＭＩＤの値は，システム内部においてユニーク

であれば，任意でよい。この命令によって，ハードウェアは，ソフトウェアのキー格納部を復号化し，その情報を権利管理テーブル１７に格納して新しいＰＭＴレコードを作成する。同時に，ＰＳＷのカレントＰＭＩＤのフィールドを，これから実行しようとする権利管理対のＰＭＩＤの値にセットする。権利管理テーブル１７のイニシャルビットおよびＰビットを＂１＂にセットする。

③　ＲＳＥ命令を実行する。カレントＰＭＴレコードは，新しく生成されたレコードであり，このイニシャルビットが＂１＂である場合に，ＲＳＥ命令が実行されると，プロセッサの状態はＰＭ状態に遷移し，許諾条件プログラムに実行制御が渡される。同時に，イニシャルビットはクリアされ，以後，オペレーティング・システムから許諾条件プログラムへ制御を移す手段は，Ｓ割込みに対するＲＳＥ命令の実行以外に存在しなくなる。

④　上記③までで，権利管理対が生成され，許諾条件プログラムの先頭から実行が開始される。許諾条件プログラムの先頭部分には，利用者との会話を行うプログラム，ソフトウェア本体の実行時間等の計量を開始するために，Ｐ割込み要求テーブルに割込み要求を書き込むプログラム等を置くことができる。その後，ソフトウェア本体に制御を移す命令を発行する。この命令では，Ｐビットのクリアが行われる。

以後，Ｐ割込みとＲＰＥ命令によって，ＵＳ状態とＰＭ状態とを間を行き来しながらプログラムの実行がなされる。権利管理対の実行を終了する場合，オペレーティング・システムは，消去したい権利管理対のＰＭＩＤをパラメータとするＰＭＴ消去命令を発行する。これにより，ハードウェアによって，権利管理テーブルの対応するレコードおよびＰＩＴの該当エントリが消去される。

権利管理に関する命令について，例えばいかなる命令を用意すればよいかについては，以上の実施例の説明で明らかであるが，各命令の機能変更，拡張は任意になし得る。

〔発明の効果〕

以上説明したように，本発明によれば，マルチプログラミングの環境において，保全性，柔軟性に優れた権利管理を行うことが可能になり，ソフトウェアを自由に大量に流通させ，かつソフトウェアの保護が充分であるシステムを構築できるようになる。

４．図面の簡単な説明

第１図は本発明の原理説明図，

第２図は本発明を用いたソフトウェアの流通形態の例を説明するための図，

第３図は本発明による権利管理の説明図，

第４図は権利管理状態の状態遷移図，

第５図は本発明に用いるハードウェア構成例，

第６図は権利管理対象となるソフトウェアの構成例，

第７図は権利管理テーブルの構成例，

第８図はＰ割込み要求テーブルの構成例，

第９図はＳ割込み制御の例，

第１０図はＳ割込みからの復帰制御の例，

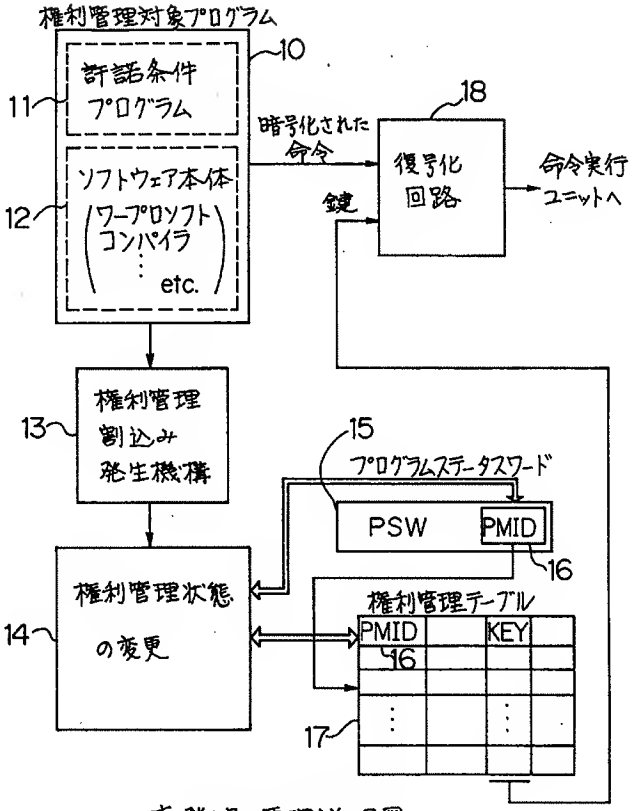
第１１図は本発明の一実施例におけるスタックの使用例，

第１２図は本発明の一実施例における空間と鍵の関係説明図を示す。

図中，１０は権利管理対象プログラム，１１は許諾条件プログラム，１２はソフトウェア本体，１３は権利管理割込み発生機構，１４は権利管理状態の変更処理，１５はプログラムステータスワード，１６は権利管理ＩＤ，１７は権利管理テーブル，１８は復号化回路を表す。

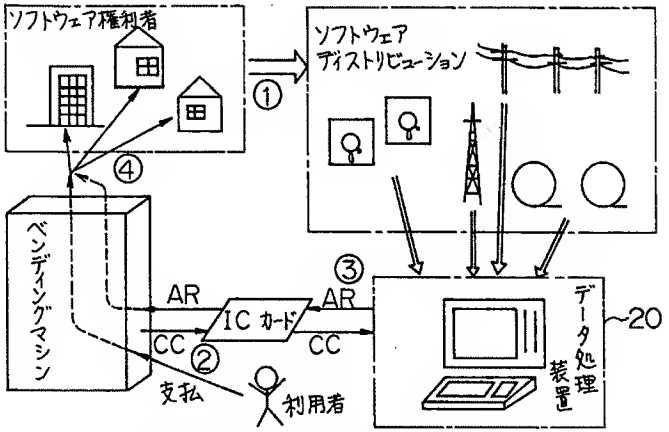
特許出願人　　森　亮一

代理人　弁理士　　小笠原　吉義（外２名）



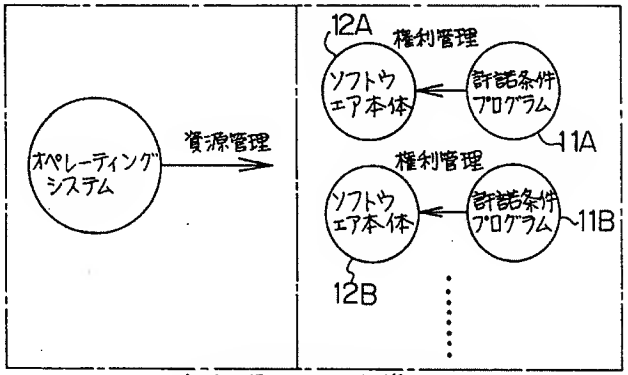
本発明の原理説明図

第 1 図



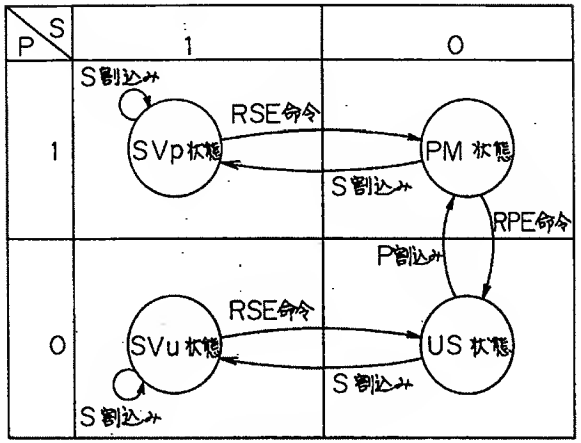
本発明を用いたソフトウェア流通形態

第 2 図



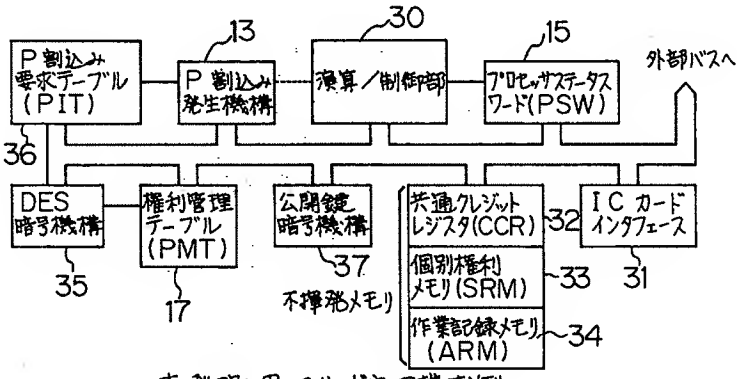
本発明による権利管理

第 3 図



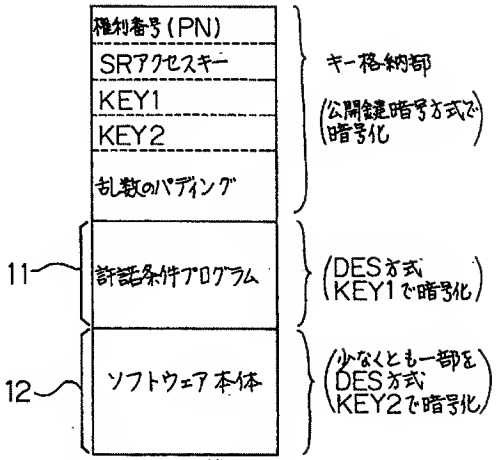
状態遷移図

第 4 図



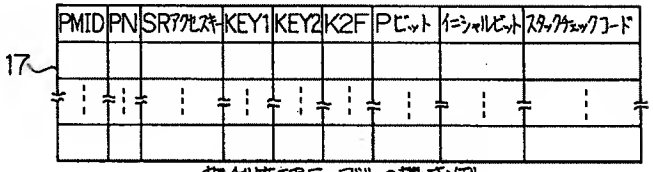
本発明に用いるハードウェア構成例

第 5 図



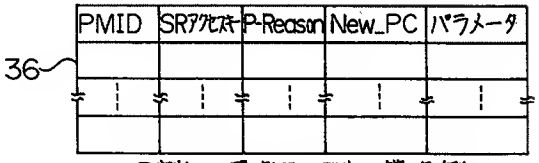
ソフトウェアの構成例

第 6 図



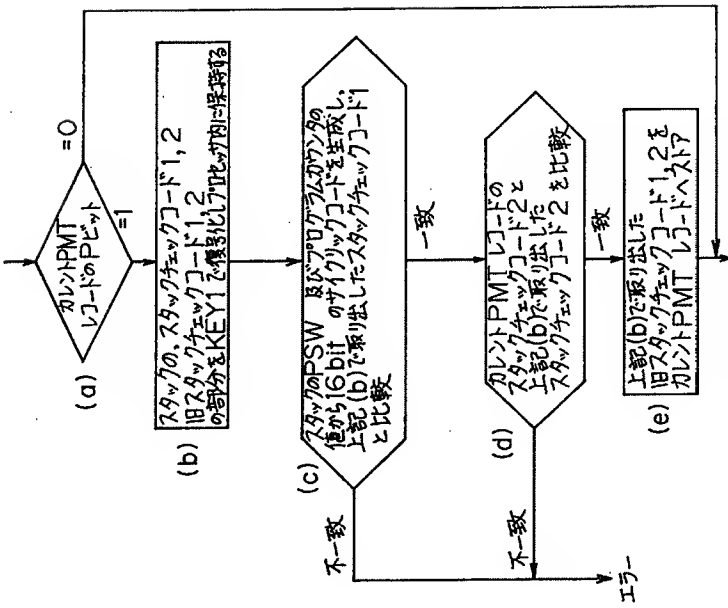
権利管理テーブルの構成例

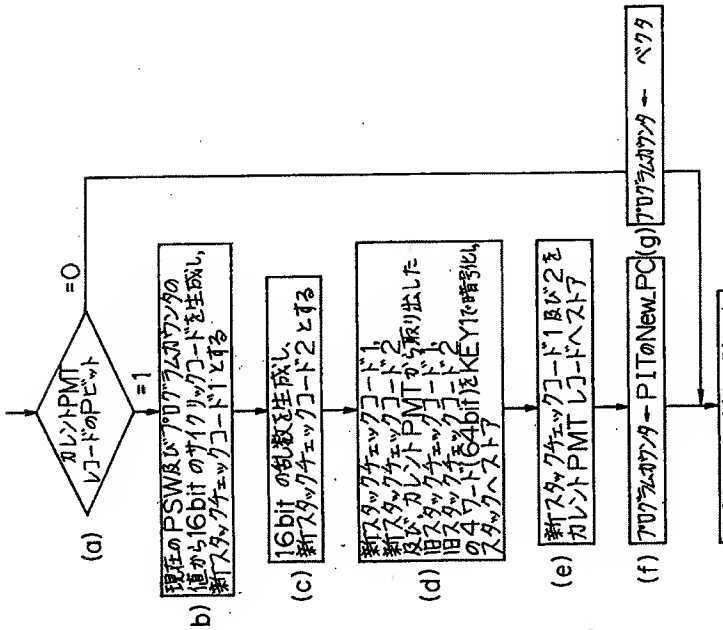
第 7 図

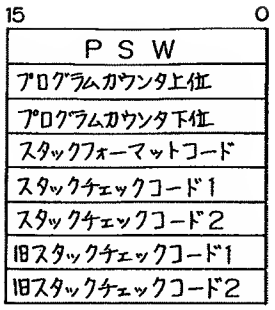


P割込み要求テーブルの構成例

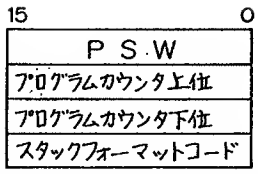
第 8 図


(a) カレントPMT レコードのPビット
=0
=1
(b) スタックの，スタックチェックコード1，2 旧スタックチェックコード1，2 の部分をKEY1で模様化プロセッサ内に保持する
(c) スタックのPSW 及びプログラムカウンタの 値から16bit のサイクリックコードを生成し， 上記(b)で取り出したスタックチェックコード1 と比較
一致
不一致
(d) カレントPMT レコードの スタックチェックコード2と 上記(b)で取り出した スタックチェックコード2 を比較
一致
不一致
エラー
(e) 上記(b)で取り出した 旧スタックチェックコード1，2を カレントPMT レコードへストア
(f) PSWのSビット ← '0'
(g) プログラムカウンタ ← スタックにストアされている プログラムカウンタ
割込み発生直前に実行されていた ルーチンへ復帰
S割込みからの復帰制御
第 10 図

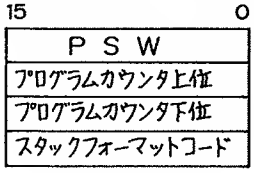

(a) カレントPMT レコードのPビット
=0
=1
(b) 現在のPSW及びプログラムカウンタの 値から16bit のサイクリックコードを生成し， 新スタックチェックコード1とする
(c) 16bit の桁数を生成し， 新スタックチェックコード2 とする
(d) 新スタックチェックコード1， 新スタックカレントPMTから取り出した 旧スタックチェックコード2 の4ワード(64bit)をKEY1で暗号化し， スタックへストア
(e) 新スタックチェックコード1及び2を カレントPMT レコードへストア
(f) プログラムカウンタ ← PITのNew.PC(g)
プログラムカウンタ ← ベクタ
(h) PSW，プログラムカウンタ， スタックフォーマットコードを スタックへストア
(i) PSWのSビット ← "1"
オペレーティングシステムの 割込み処理ルーチンへ
S割込み処理ルーチン
第 9 図



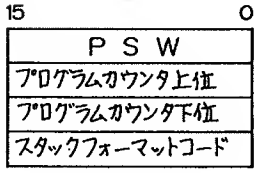
(a) PM状態時におけるS割込み発生時



(b) US状態時におけるS割込み発生時



(c) SV状態時におけるS割込み発生時



(d) US状態時におけるP割込み発生時

スタックの使用例

第 11 図

	SV状態	PM状態	US状態
空間	スーパバイザ空間	ユーザ空間	
鍵	—	カレントPMTのKEY1	カレントPMTのKEY2

空間と鍵の関係

第 12 図

PATENT ABSTRACTS OF JAPAN

(11)Publication number : 64-068835
(43)Date of publication of application : 14.03.1989

(51)Int.Cl. G06F 9/06

(21)Application number : 62-227106
(22)Date of filing : 10.09.1987

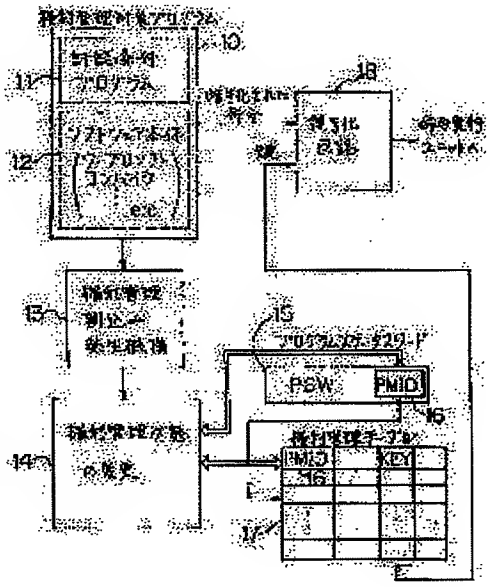
(71)Applicant : MORI RYOICHI
(72)Inventor : MORI RYOICHI
TASHIRO SHUICHI

(54) SOFTWARE RIGHT MANAGEMENT CONTROL METHOD

(57)Abstract:

PURPOSE: To perform the right management superior in maintainability and flexibility in the environment of multiprogramming by individually assigning the key, with which ciphered instructions are deciphered, to a program or a program group as the right management object and switching the deciphering keys by a right management interrupt.

CONSTITUTION: A right management interrupt generating mechanism 13 which controls the change related to the right management state is provided independently of the supervisor interrupt which starts the processing function of an operating system or the like. When the right management interrupt is generated by the right management interrupt generating mechanism 13, the right management state is saved to a right management table 17 and a current right management ID 16 of a program status word 15 is changed by a right management state change processing 14, and the key with which instructions are deciphered is switched. Thus, right management in the environment of multiprogramming is possible.



Partial Translation of JPA 64-068835

<Page 3, upper right column line 19-lower left column line 2>

A program 10 to be right managed is, for example, comprised by a pair of a permission requirement program 11 and a software body 12.

<Page 4, upper right column line 8-14>

① A software right holder who develops a software, when the software is distributed, generates a permission requirement program for describing proviso for requesting a software user, combines the generated permission requirement program with the developed software, and encrypts the combined data and distributes it via any medium including broadcast by electronic wave.

<Page 5, upper right column line 17-lower left column line 5>

As shown in fig. 3, an operating system executes a resource management in a supervisor status, while each permission requirement program 11A, 11B, and so on for a program to be right managed executes a right management by monitoring execution of software 12A, 12B and so on in each right management status. Hereinafter, the permission requirement program and software

controlled by it are called as a pair of right management.

<Page 6, upper left column lines 16-19>

A structure of software implemented in an apparatus as shown in fig. 5, for example, is shown in fig. 6. The structure consists of three portions, a key storage portion, a permission requirement program 11 and a software 12.

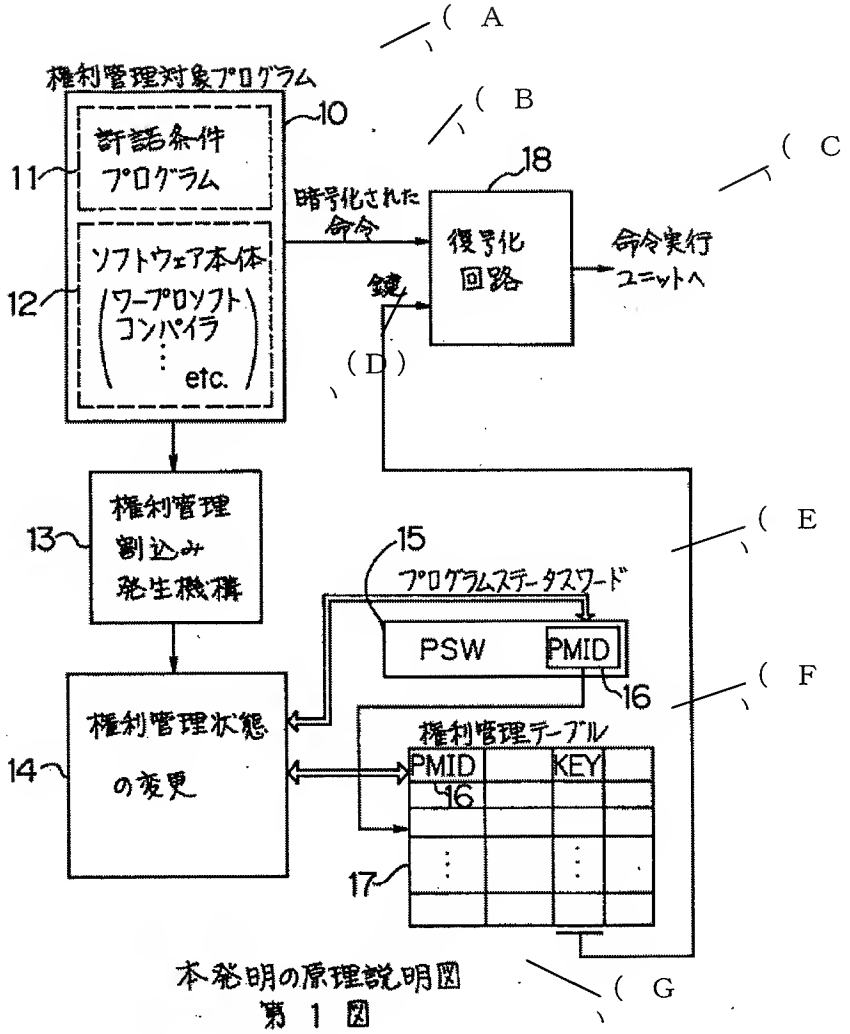


Fig. 1

(A) program to be right managed

(B) encrypted instruction

(C) instruction execution unit

(D) key

(E) program status word

(F) right management table

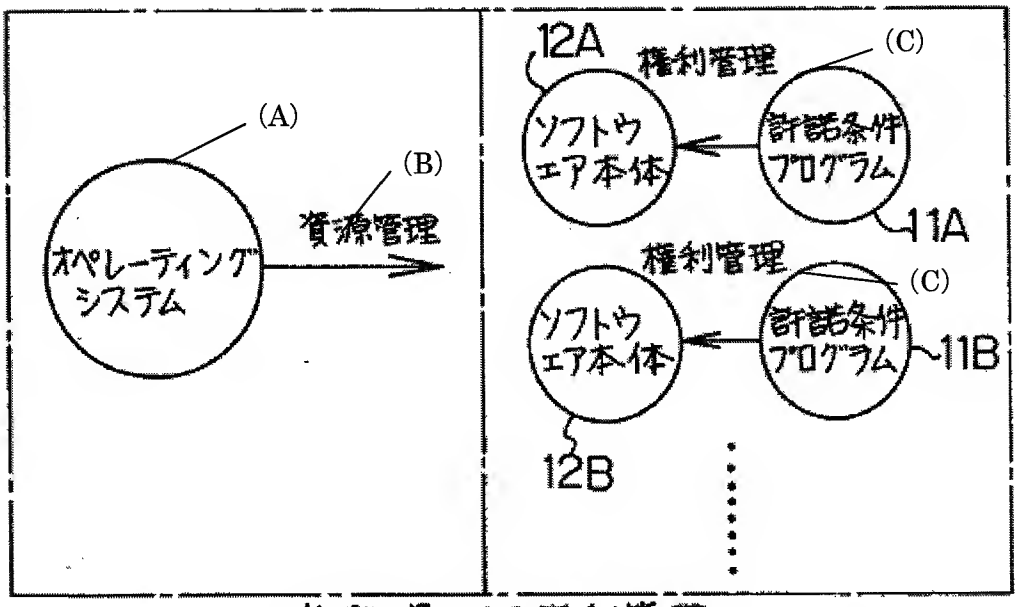
(G) principle explanation figure of present invention

11 permission requirement program

12 software (word processing software, compiler, etc.)

13 right management interruption occurrence mechanism

14 change of right management status



第 3 図

Fig. 3

(A) operating system

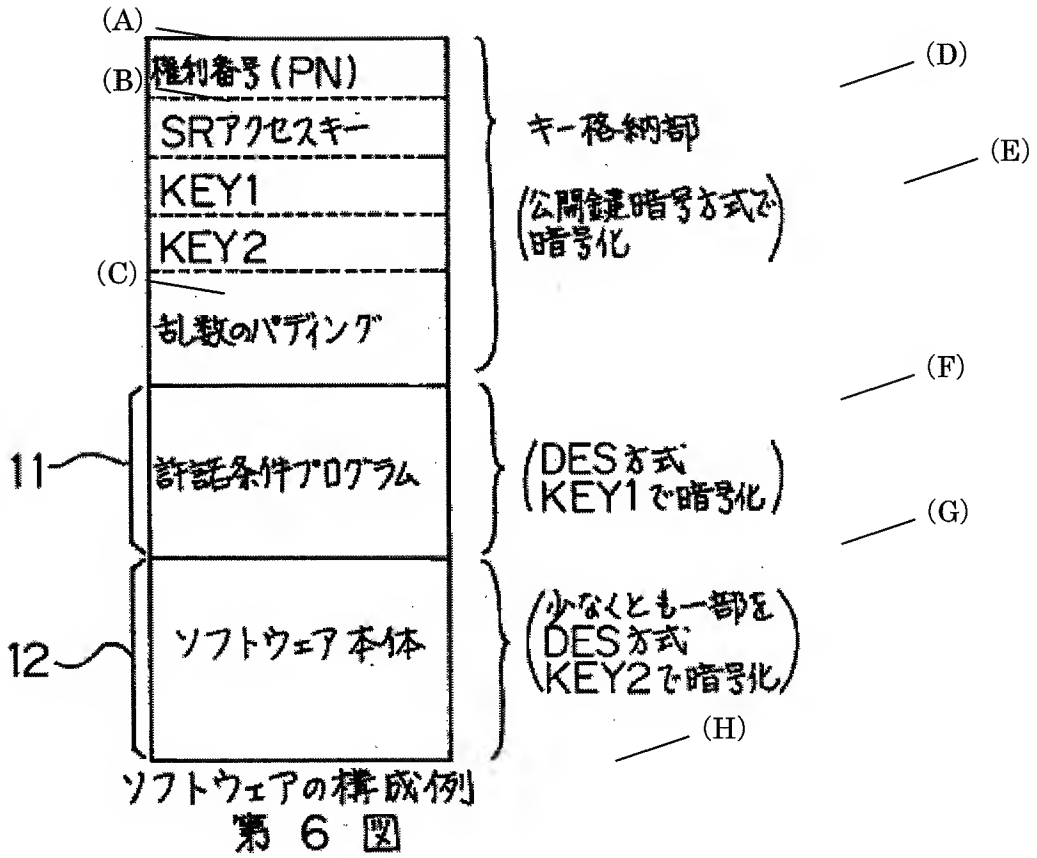
(B) resource management

(C) right management

(D) right management according the present invention

11A and 11B permission requirement program

12A and 12B software



ソフトウェアの構成例

第 6 図

Fig. 6

(A) right number (PN)

(B) SR access key

(C) Padding of random number

(D) Key storage portion

(E) Encryption by public key cryptosystem

(F) DES scheme, encryption with KEY1

(G) At least partial portion with DES scheme, encryption with KEY2

(H) Structure example of software

11 permission requirement program

12 software